Midterm 3 Study Guide

This is not meant to be an exhaustive list of everything you need to know, but it is a good starting place.

1. What is the formal definition of a **Turing machine**?
2. Describe what a **configuration** for a Turing machine is.
3. What are the possible outcomes of running a Turing machine on some input?
4. When do we say a language is Turing-recognizable?
5. When do we say that a language is decidable?
6. Describe in fairly low detail Turing machines that decide "simple" problems (e.g. some regular languages or CFGs).
7. What is a multi-tape Turing machine?
8. Describe how we can simulate a multi-tape Turing machine with a single-tape Turing machine.
9. What is a non-deterministic Turing machine?
10. Describe how we can simulate a non-deterministic Turing machine via a multi-tape Turing machine.
11. Describe how the string representation of a graph might be.
12. Describe how the string representation of a Turing machine might be.
13. Describe the acceptance, empty language and equivalent languages problems for DFAs, and show that the corresponding languages are decidable.
14. Describe the acceptance, empty language and equivalent languages problems for CFGs. Show that the acceptance and empty language problems are decidable.
15. Describe the languages A_{TM} and HALT_{TM}. Show that they are undecidable (this is effectively the Halting Problem question).
16. Show that if a language is both Turing-recognizable and co-Turing-recognizable, then it is decidable.
17. Show that E_{TM} and EQ_{TM} are undecidable.
18. Define the concept of mapping reducibility and computable functions.
19. If $A \leq_m B$, what statements can be made about when the decidability/undecidablity/Turing-recognizability/non-Turing-recognizability and so on of one language implies the same for the other?
20. Show that EQ_{TM} is neither Turing-recognizable and co-Turing-recognizable.

21. How do we define the time complexity of a decidable Turing-machine?

22. What is the time complexity class TIME$(t(n))$.

23. What is the class P? Show examples of languages that are in P.

24. What is the class NP? Show examples of languages that are in NP.

25. Define polynomial-time reducibility.

26. What problems are called NP-complete?

27. Describe the problems SAT, 3SAT, CLIQUE, VERTEX-COVER, HAMPATH, 3COLOR, SUBSET-SUM.

28. State the Cook-Levin theorem. Explain its significance.

29. Show that 3SAT is polynomial time reducible to CLIQUE as well as to VERTEX-COVER. Use this to show these are NP-complete.